PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-036809
(43)Date of publication of application : 02.02.2000

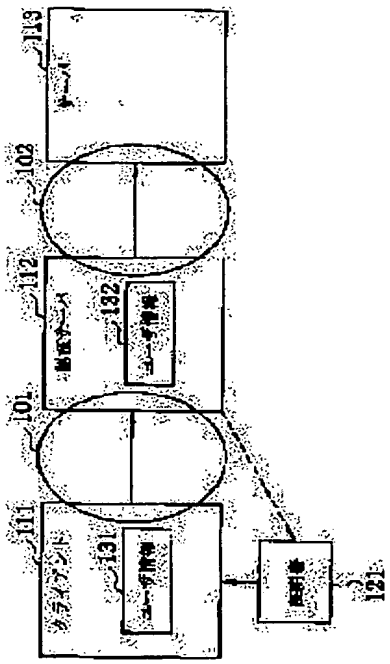
(51)Int.Cl. H04L 9/32
G06F 15/00
H04L 9/08

(21)Application number : 10-203654
(22)Date of filing : 17.07.1998

(71)Applicant : HITACHI LTD
(72)Inventor : HIDAKA TAIJI
SAITO YOKO

(54) METHOD FOR SIMPLY AUTHENTICATING USER AND RECORD MEDIUM WITH ITS PROGRAM STORED THEREIN

(57)Abstract:
PROBLEM TO BE SOLVED: To relieve the overhead of an authentication server by applying simple authentication at the next time with respect to a user that has been authenticated to conduct authentication in a short time without degrading a security level.
SOLUTION: When a client 111 makes a request of connection with respect to a server 113 to an authentication server 112, the authentication server 112 authenticates the user, based on identification information and authentication information. Furthermore, the server 112 stores sets of user information 131, 132 consisting of addresses of transmission source and destination of the user, a port number and a session key generated at the authentication and stores number of connections in use applied by the user at authentication or connection information for a connection existence time, and conducts simple authentication of the user based on the connection information and the user information, when it is required to authenticate the user again. The authentication by using a certificate 121 is conducted only for a first connection request.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号
特開2000－36809
（Ｐ２０００－３６８０９Ａ）
(43)公開日　平成12年２月２日(2000.2.2)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード（参考）
H04L 9/32		H04L 9/00	673 A	5B085
G06F 15/00	330	G06F 15/00	330 B	5K013
H04L 9/08		H04L 9/00	601 C	
			601 E	

審査請求　未請求　請求項の数４　ＯＬ　（全11頁）

(21)出願番号　特願平10－203654

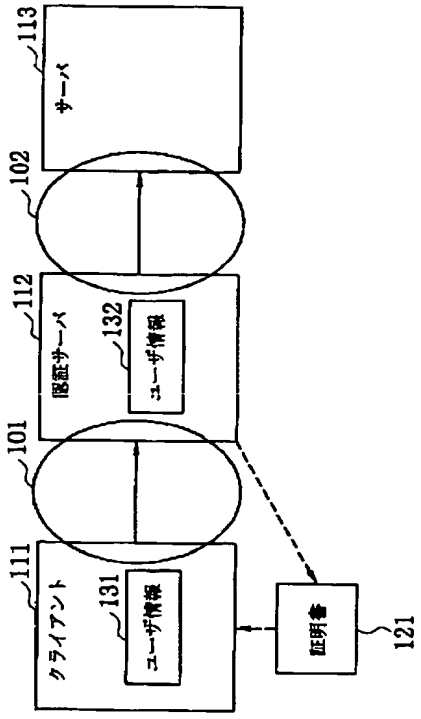
(22)出願日　平成10年７月17日(1998.7.17)

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　日▲高▼　大治
神奈川県横浜市戸塚区戸塚町5030番地　株式会社日立製作所ソフトウェア開発本部内
(72)発明者　齋藤　洋子
神奈川県横浜市戸塚区戸塚町5030番地　株式会社日立製作所ソフトウェア開発本部内
(74)代理人　100077274
弁理士　磯村　雅俊　（外１名）

Ｆターム(参考)　5B085 AE06 AE09 AE13 AE23 AE29 BG07
5K013 AA02 BA02 GA00 GA03

(54)【発明の名称】ユーザの簡易認証方法およびそのプログラムを格納した記録媒体

(57)【要約】
【課題】認証済みのユーザに対しては次回は簡易認証を行い、セキュリティレベルを落とさずに短時間で認証を行い、認証サーバのオーバーヘッドを低減する。
【解決手段】クライアント111から認証サーバ112に対して、サーバ113とのコネクションの接続要求を行うと、認証サーバ112はユーザを識別情報と認証情報により認証し、同時にユーザの送信元先のアドレス、ポート番号と認証時に生成されるセション鍵から構成されるユーザ情報131,132を記憶し、認証時にユーザが申請した使用コネクション数あるいはコネクション存続時間のコネクション情報を記憶し、再度ユーザを認証する必要がある時には、前記コネクション情報あるいはユーザ情報をもとにユーザを簡易認証する。証明書121による認証は、最初の接続要求時のみ行う。



1

【特許請求の範囲】
【請求項１】　ネットワークシステムに接続されたクライアントからサーバへのコネクション接続要求時に、認証サーバが該クライアントのユーザを認証する認証方法において、
該認証サーバは、該クライアントがサーバにコネクション接続要求する時に、ユーザを識別情報と認証情報により認証するステップと、
該ユーザの送信元アドレス、ポート番号と、認証時に生成されるセション鍵、および前記認証時にユーザが申請した使用コネクション数あるいはコネクション存続時間から構成されるユーザ情報を記憶するステップと、
再度、上記ユーザを認証する必要がある時には、前記ユーザ情報をもとにユーザを簡易認証するステップとを有することを特徴とするユーザの簡易認証方法。
【請求項２】　前記認証サーバは、再度、ユーザを認証する時に、前記記憶されたユーザ情報と今回のユーザの送信元アドレスとポート番号が一致するか否かを確認するステップを有し、
前記クライアント側では、ユーザが乱数を生成し、セション鍵により暗号化するステップと、
該乱数及び暗号化された該乱数を該認証サーバに送信するステップと、
該乱数と暗号化された該乱数とから次のセション鍵を生成するステップと該セション鍵を記憶するステップとを有することを特徴とする請求項１に記載のユーザの簡易認証方法。
【請求項３】　前記認証サーバは、再度、ユーザを認証する時に、ユーザによるコネクション接続要求が前記記憶されたユーザ情報に符合するか否かを確認するステップと、
クライアントにより発生させた乱数と、該乱数をセション鍵により暗号化されたものを受信するステップと、
暗号化された該乱数をセション鍵により復号するステップと、
復号化された乱数と受信した乱数を比較するステップと、
該乱数と暗号化された該乱数とから次のセション鍵を生成するステップと、
該セション鍵を記憶するステップとを有することを特徴とするする請求項１に記載のユーザの簡易認証方法。
【請求項４】　請求項１から請求項３までのいずれかに記載のユーザ簡易認証方法の各動作を、プログラムに変換し、変換されたプログラムを格納したことを特徴とする記録媒体。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、ネットワークシステムにおけるユーザ認証処理の、特にモバイル環境におけるオーバーヘッドを解決するためのユーザの簡易認証方法に関する。

2

【０００２】
【従来の技術】従来、暗号化通信を行うためには、特開平9-148993号公報に記載された「通信システム」に示されるように、クライアントからサーバに対してコネクション要求があると、クライアントは認証サーバに対して使用する暗号アルゴリズムの種別を決定するためにネゴシエーションを行っていた。ユーザ認証は、ネゴシエーション時に行われている。そのため１つのクライアントから同じサーバに対して複数のコネクションを張るような場合でも、各コネクションごとに認証処理が必要であった。このように、インターネットの普及に伴い、不特定多数の人が必要に応じてネットワークシステムにアクセスできるようになってきた。そのため、クライアントがサーバに対してコネクション要求してきた場合には、クライアントのユーザを認証することが重要である。しかしながら、実際の運用を考えると、ネットワークシステムが大規模化するに伴って、通信のコネクションが張られる度毎に行わなければならない認証処理のオーバーヘッドが問題になる。例えば、Webブラウザなどで1つのアプリケーションサーバに複数のコネクションを張るような場合、最初のコネクションでユーザ認証を行い、認証されればあとのコネクションでは簡易認証だけを行うユーザ要求ができれば極めて便利である。特に、モバイル環境に適用可能な認証方法の提案が課題となっている。ある程度のセキュリティを確保しながら認証処理のオーバヘットを削減する技術の検討が必要になった。
【０００３】
【発明が解決しようとする課題】前述のように、ネットワークシステムが大規模化すると、通信のコネクションが張られる度毎にユーザの認証を行う処理がサーバ側のオーバーヘッドとなっていた。１つのアプリケーションサーバに複数のコネクションを張るような場合、１回目のコネクションでユーザ認証を行い、認証されれば２回目以降のコネクションでは簡易認証だけを行うユーザ要求ができればオーバーヘッドも低減されると考えられる。特に、モバイル環境では、多数のユーザが移動中にサーバにアクセスするため、その都度、ユーザの認証を行うのではサーバ側の負荷は過大になる。その結果として、特にモバイル環境に適用可能な認証方法の実現を希望しており、ある程度のセキュリティを確保しながら認証処理のオーバヘットを削減する技術の検討が必要になった。そこで、本発明の目的は、このような従来の課題を解決し、ユーザの送信元アドレスとポート番号が一致し、かつユーザのコネクション接続要求がユーザの申請に符合した場合には、セキュリティレベルを落すことなく、簡易な認証を行うことができ、短時間でのデータ通信を可能にするユーザの簡易認証方法およびそのプログラムを格納した記録媒体を提供することにある。
【０００４】

【課題を解決するための手段】上記目的を達成するため、本発明のユーザの簡易認証方法では、①クライアントがサーバにコネクション接続要求する時に、ユーザを識別情報と認証情報により認証するステップと、ユーザの送信元のアドレス、ポート番号と認証時に生成されるセション鍵、および前記認証時にユーザが申請した使用コネクション数あるいはコネクション存続時間から構成されるユーザ情報を記憶するステップと、再度ユーザを認証する必要がある時には前記コネクション情報あるいはユーザ情報をもとにユーザを簡易認証するステップとからなることを特徴とする（図３の通常認証の後、図４の簡易認証の動作参照）。

【０００５】特に、②ユーザの送信元アドレスとポート番号が前回の認証時と一致し、かつユーザのコネクション接続要求がユーザの申請に符合した場合について、２回目以降の認証処理を簡易化する。具体的には、記憶されたユーザ情報と今回のユーザの送信元アドレスとポート番号が一致するか否かを確認するステップと、ユーザが乱数を生成しセション鍵により暗号するステップと、前記乱数及び前記暗号化された乱数を認証サーバに送信するステップと、前記乱数と前記暗号化された乱数とから次のセション鍵を生成するステップと、前記セション鍵を記憶するステップとからなることを特徴としている（図７の確認動作と図４のユーザ側動作参照）。また、③ユーザによるコネクション接続要求が前記記憶されたユーザ情報に符合するかどうか確認するステップと、認証サーバがクライアントにより発生させた乱数と、前記乱数をセション鍵により暗号化されたものを受信するステップと、前記乱数と前記暗号化された乱数をセション鍵により復号するステップと、前記復号化された乱数と前記受信した乱数を比較するステップと、前記暗号化された乱数とから次のセション鍵を生成するステップと、前記セション鍵を記憶するステップとからなることを特徴としている（図４のサーバ側動作参照）。さらに、本発明によるプログラム格納の記録媒体は、④上記①～③に記載の各動作ステップ群をプログラムに変換し、変換されたプログラムを記録媒体に格納することを特徴としている。

【０００６】

【発明の実施の形態】以下、本発明の実施例を、図面により詳細に説明する。図１は、本発明が適用されるネットワークシステムの構成図である。図１において、クライアント111とサーバ113は、それぞれ独立したネットワーク101及び102を介して接続されている。クライアント111及びサーバ113は複数個存在するが、ここでは簡単のために1個ずつしか示されていない。また、認証サーバ112は、ネットワーク101と102を中継する位置で、前記２つのネットワーク間のコネクションを管理し、クライアント111に対しては事前に証明書121を発行しておく。前記ネットワーク101及び102間では、秘密鍵暗号方式を使用してコネクションごとに生成したセション鍵を用いて暗号通信する。クライアント111には、そのクライアント端末を共用するユーザが複数存在し、それらのユーザ情報131が内部メモリに格納されている。一方、認証サーバ112の内部メモリにも、各クライアント端末111からの接続要求時に送られるユーザ情報を読み取ったものが格納される。前記認証サーバ112から事前に配布されている証明書121の用紙は、規定のフォーマットが記載されており、各ユーザは、クライアント111の内部メモリに格納されたユーザ情報を読み込み、それを証明書121に書き込むことにより、証明書を完成させる。完成された証明書121には、前記セション鍵を生成するための共有鍵情報に加えて、クライアント111のユーザのユーザ名、当証明書の有効期限、接続サーバを限定するためのアクセス制御情報が格納されている。

【０００７】図２は、図１におけるユーザ情報の構成図である。本発明では、図１におけるユーザ情報131,132を用いることにより、クライアント111からサーバ113への再度の接続要求に対して認証処理を簡易化することを目的としている。前記ユーザ情報131,132は、接続先あるいは接続要求元のアドレスとポート番号、セション鍵、コネクション数による制御か、あるいはコネクション存続時間による制御かを示すフラグ、使用するコネクション数あるいはコネクション存続時間、及び次のユーザ情報へのポインタ情報から構成される。複数のユーザ情報201,202,203は、図２に示すようにリストの先頭からリストの最後に至るまでそれぞれポインタにより結合される。ユーザ情報131,132は、コネクションごとにクライアント111側と認証サーバ112側で生成される。例えば、１つのコネクションについて、クライアント111側のユーザ情報131には接続先であるサーバ113のアドレスとポート番号が格納され、認証サーバ112側のユーザ情報132には接続要求元であるクライアント111のアドレスとポート番号が格納されている。また、使用するコネクション数による制御とコネクション存続時間による制御のどちらを採用するかというネゴシェーションについては、事前にシステム定義で指定しておく運用もあるが、本実施例ではクライアント111からサーバ113への接続要求の最初の通常認証処理時に設定している。クライアント111がユーザ情報131に格納されているアドレスとは異なるアドレスのサーバへアクセスした場合、また、認証サーバ112がユーザ情報132に格納されているアドレスとは異なるアドレスのクライアントからの認証を行った場合、新たなユーザ情報が生成されるが、これは図２に示すようにリストとして格納される。

【０００８】このようなネットワーク環境において、クライアント111の端末には、ＣＰＵが配置され、サーバ113に接続してデータを送受信するアプリケーションを実行することにより動作するようになっている。クライアント111からサーバ113に接続要求があった場合、クライ

アント111はまず認証サーバ112と接続し、ユーザ認証を行う。認証が正しく行われた場合、セション鍵が生成され、以降の通信で使用するために前記セション鍵をクライアント111と認証サーバ112とで共有しておく。その後、このセション鍵を使用してクライアント111は認証サーバ112間と暗号通信を行い、認証サーバ112が中継することによりサーバ113に接続する。以上は、従来からの処理を説明したものである。

【０００９】図３は、本発明で最初に行われる通常の認証処理のフローチャートであり、図４は、本発明の一実施例を示す簡易認証処理のフローチャートである。図３および図４により、クライアント、認証サーバ、サーバとの間の処理シーケンスを説明する。図３では、ユーザからのサーバ接続要求を受けて(ステップ301)、クライアント111で認証サーバ112に対して通常認証要求を行う(ステップ302)。クライアント111が配布されている証明書を認証サーバ112に送信すると(ステップ303)、認証サーバ112では、証明書を受信した後(ステップ305)、自分が保持している証明書と比較して認証を行う(ステップ306)。一致しなかった場合、認証失敗をクライアント111に通知し、クライアント111との接続を切断する(ステップ317)。これに対して一致した場合には、クライアント111に認証成功の応答を送信し(ステップ307)、セション鍵を生成する(ステップ309)。認証が成功すると(ステップ308)、クライアント111側では、セション鍵を生成し(ステップ310)、ユーザ情報131としてアドレス、ポート番号、セション鍵、フラグ、コネクション数あるいはコネクション存続時間等の情報を設定し、認証サーバ112に送信する(ステップ311)。そして、ユーザ情報131をリストに追加する(ステップ313)。なお、当設定の詳細のフローチャートについては図５で説明する。また、図３では、認証処理(ステップ306)を簡単に記載しているが、実際にはさらに乱数のやり取り等の複数のシーケンスが存在するが、それについては図８で後述する。次に、認証サーバ112側では、受信した前記情報から図６に示すユーザ情報132の設定処理(ステップ312,314)を行い、サーバ113に対してクライアント111からの接続要求を中継する(ステップ315)。サーバ113との通信が終了したならば(ステップ316)、クライアント側と認証サーバ側とで接続断とする(ステップ318,317)。

【００１０】図８は、図３における通常認証時の認証処理の詳細なシーケンスチャートである。図３において、クライアント側は配布された証明書を送信した後(ステップ303)、図８に移り、乱数を発生して、その乱数を認証サーバ側に送信する(ステップ801)。認証サーバ側では、受信した乱数を自身が保存している暗号鍵で暗号化し、クライアント側に送信する(ステップ802)。クライアント側では、暗号化したものを受信し(ステップ803)、乱数を復号して(ステップ804)、自身が保存している乱数と復号した乱数とを比較する(ステップ805)。その結果、一致していれば、暗号化したものを再度、認証サーバ側に送信する(ステップ806)。認証サーバ側では、暗号化したものを受信して(ステップ807)、これと最初にステップ802で暗号化したものとを比較し、一致するとともに、クライアント側から送られた証明書の内容を自身が保存している内容と比較して、一致すれば、認証成功と判断し、認証成功をクライアント側に送信する(ステップ307)。クライアント側では、認証成功を受信すると(ステップ308)、セション鍵を生成して(ステップ310)、以後はセション鍵で暗号化して通信を行う。

【００１１】次に、図４により本発明における簡易認証処理について説明する。ユーザからのサーバ接続要求を受けた時(ステップ401)、クライアント111では、接続先のサーバのアドレスとポート番号を前記格納されたユーザ情報131と比較し、一致したならば、この接続先への認証は終わっているものとして、認証サーバ112に簡易認証要求を送信する(ステップ402)。一方、認証サーバ112は、クライアント111からの簡易認証要求を受信した時(ステップ403)、ユーザ情報確認処理を行う(ステップ404)。当該確認処理の詳細については、図７において後述する。認証サーバ112では、それが通常認証なのか簡易認証なのかを判別するために、接続元クライアントのアドレスとポート番号を、自身が保持しているユーザ情報132と比較し、一致しなかった場合、認証失敗をクライアント111に通知し、クライアント111との接続を切断する(ステップ418)。

【００１２】一致した場合には、クライアント111は乱数を発生させ、それをクライアント111が保持しているユーザ情報131のセション鍵で暗号化し(ステップ405)、それら乱数と暗号化された乱数とを認証サーバ112に送信する(ステップ406)。認証サーバ112はそれを受信した後(ステップ407)、自身が保持しているユーザ情報132のセション鍵で、暗号化されている乱数を復号し(ステップ408)、同時に送られてくる乱数と比較する(ステップ409)。一致しなければ、認証失敗をクライアント111に通知し、クライアント111との接続を切断する(ステップ418)。一致すれば、認証成功をクライアント111に送信する(ステップ410)。クライアント111では、認証成功を受信すると(ステップ411)、乱数と暗号化された乱数から新たなセション鍵を生成し(ステップ413)、新たなセション鍵をユーザ情報131に保存する(ステップ415)。一方、認証サーバ112も、クライアント111と同様にしてセション鍵を生成し(ステップ412)、ユーザ情報132に保存する(ステップ414)。次に、認証サーバ112は、クライアント111からの接続要求をサーバ113に中継する(ステップ416)。サーバ113との通信が終了した時点で(ステップ417)、クライアント側と認証サーバ側との接続を断にする(ステップ419,418)。

【００１３】以上が、2回目以降に行われる本発明の簡易認証処理である。図３および図８に示した通常認証処

理と、図４に示した本発明の簡易認証処理とを比較すると明らかなように、通常認証では証明書を送受信しているのに対して、簡易認証では証明書は省略して簡易認証要求のみで済むので、通信回数が１回少なくなり、また、通常認証ではクライアント側で乱数を発生して乱数自身の送受信を行い、クライアント側で乱数比較を行うので、乱数の通信回数が３回必要であるのに対して、簡易認証ではクライアント側で乱数を発生するがその暗号化したものを認証サーバに送信して、認証サーバ側で乱数の比較を行うので、通信回数は１回で済み、さらに通常認証ではユーザ情報の有効期限情報を送受信するのに対し、簡易認証ではユーザ情報確認処理（ステップ404）の時点でユーザ情報の有効期限情報を送受信しているので、余分な通信回数は不要となる。結局、証明書の送信で1回、乱数比較のための送信で2回、ユーザ情報の有効期限の送信で1回の合計4回も通信回数が減少する。その結果、短時間でユーザ認証を行うことができるため、認証サーバのオーバーヘッドも低減される。また、簡易認証について生成した乱数の暗号化したものを次のセション鍵として使用するため、セキュリティレベルを落とすことなく短時間での認証を実現することが可能である。

【００１４】図５は、クライアントにおけるユーザ情報設定処理のフローチャートである。図５により、クライアント111側でユーザ情報131を設定する処理の手順について説明する。クライアント111が通常認証に成功した場合には、クライアント111から認証サーバ112に対して使用するコネクション数あるいは接続先へのコネクション存続時間のどちらかを申請する(ステップ502、504)。クライアントが使用するコネクション数を申請した場合、申請した数のコネクションが張られるまでユーザ情報を有効とする(ステップ503)。また、クライアントが接続先へのコネクション存続時間を申請した場合、当該接続先サーバへのコネクションが張られている間、ユーザ情報を有効とする(ステップ505)。その後、接続先サーバのアドレス、ポート番号、生成されたセション鍵、有効期限をユーザ情報としてリストに追加する(ステップ506)。

【００１５】図６は、認証サーバにおけるユーザ情報設定処理のフローチャートである。図６により、認証サーバ112でユーザ情報132を設定する処理手順について説明する。通常認証に成功した場合には、認証サーバ112はクライアント111からユーザ情報有効期限情報を受信し(ステップ601)、送られてきた情報を基にユーザ情報132のコネクション数またはコネクション存続時間のメンバを有効にする(ステップ603、604)。それから、接続元クライアントのアドレス、ポート番号、生成されたセション鍵、有効期限をユーザ情報としてリストに追加する(ステップ605)。

【００１６】図７は、認証サーバでのユーザ情報確認処理のフローチャートである。最後に図７により認証サーバ112でのユーザ情報の確認方法について説明する。簡易認証の場合には、認証サーバ112で以前認証したユーザと接続元が同じかどうかを確認(アドレスとポート番号の存在)した後(ステップ701)、ユーザ情報132の内容が有効期限内であるか否かを確認する必要がある(ステップ702,703,704)。認証サーバ112では、簡易認証時に有効期限が切れていた場合(ステップ703)、あるいは簡易認証に失敗した場合(ステップ707)、当該ユーザ情報をリストから削除する(ステップ705)。クライアント111では、簡易認証に失敗した場合、当該ユーザ情報をリストから削除する。

【００１７】図４に示す簡易認証時のシーケンスチャート、図５に示すクライアントのユーザ情報設定処理のフローチャート、図６に示す認証サーバのユーザ情報設定処理のフローチャート、図７に示すユーザ情報確認処理のフローチャート、図３に示す通常認証時のシーケンスチャート、および図８に示す通常認証時の乱数比較に際してのシーケンスチャートの各動作をプログラムに変換し、変換されたプログラムをＣＤ－ＲＯＭやハードディスク等の記録媒体に格納することにより、それらの記録媒体を通信システム中の任意のサーバやクライアント端末にローディングしてプログラムを実行すれば、本発明を任意の場所で実現することができる。

【００１８】

【発明の効果】以上説明したように、本発明によれば、ユーザの送信元アドレスとポート番号が前回の認証時と一致し、かつユーザのコネクション接続要求がユーザの申請に符合した場合について、２回目以降の認証処理を簡易化することができる。そして、簡易認証について生成した乱数の暗号化したものを次のセション鍵として使用するため、セキュリティレベルを落とすことなく短時間での認証を実現することが可能である。

【図面の簡単な説明】

【図１】本発明が適用されるネットワークシステムの構成図である。

【図２】図１におけるユーザ情報の構成図である。

【図３】通常認証時のクライアントと認証サーバとの間のシーケンスチャートである。

【図４】本発明における簡易認証時のクライアントと認証サーバ間のシーケンスチャートである。

【図５】クライアントにおけるユーザ情報設定処理のフローチャートである。

【図６】認証サーバにおけるユーザ情報設定処理のフローチャートである。

【図７】認証サーバでのユーザ情報確認処理のフローチャートである。

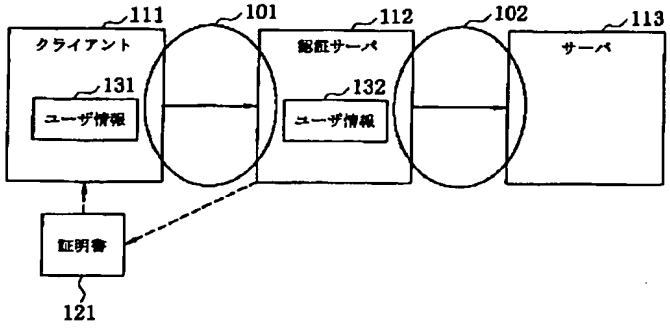
【図８】通常認証時の乱数比較に際しての詳細なシーケンスチャートである。

【符号の説明】

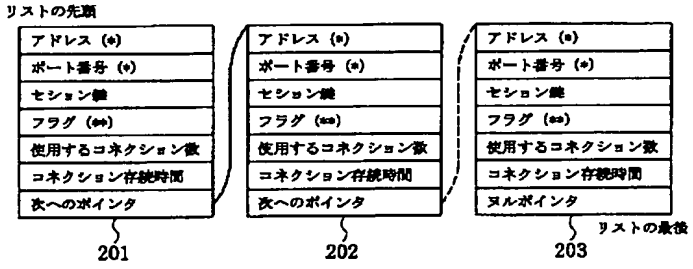
101、102：それぞれ独立したネットワーク、

111：クライアントアプリケーションが動作する計算機、
112：クライアントを認証し、ネットワーク間を中継する計算機、
113：サーバアプリケーションが動作する計算機、
121：認証サーバからクライアントへ配布された証明書、
131：クライアントにおけるユーザ情報、
132：認証サーバにおけるユーザ情報、
201, 202, 203：ポインタで結合されたユーザ情報のリスト。

【図１】

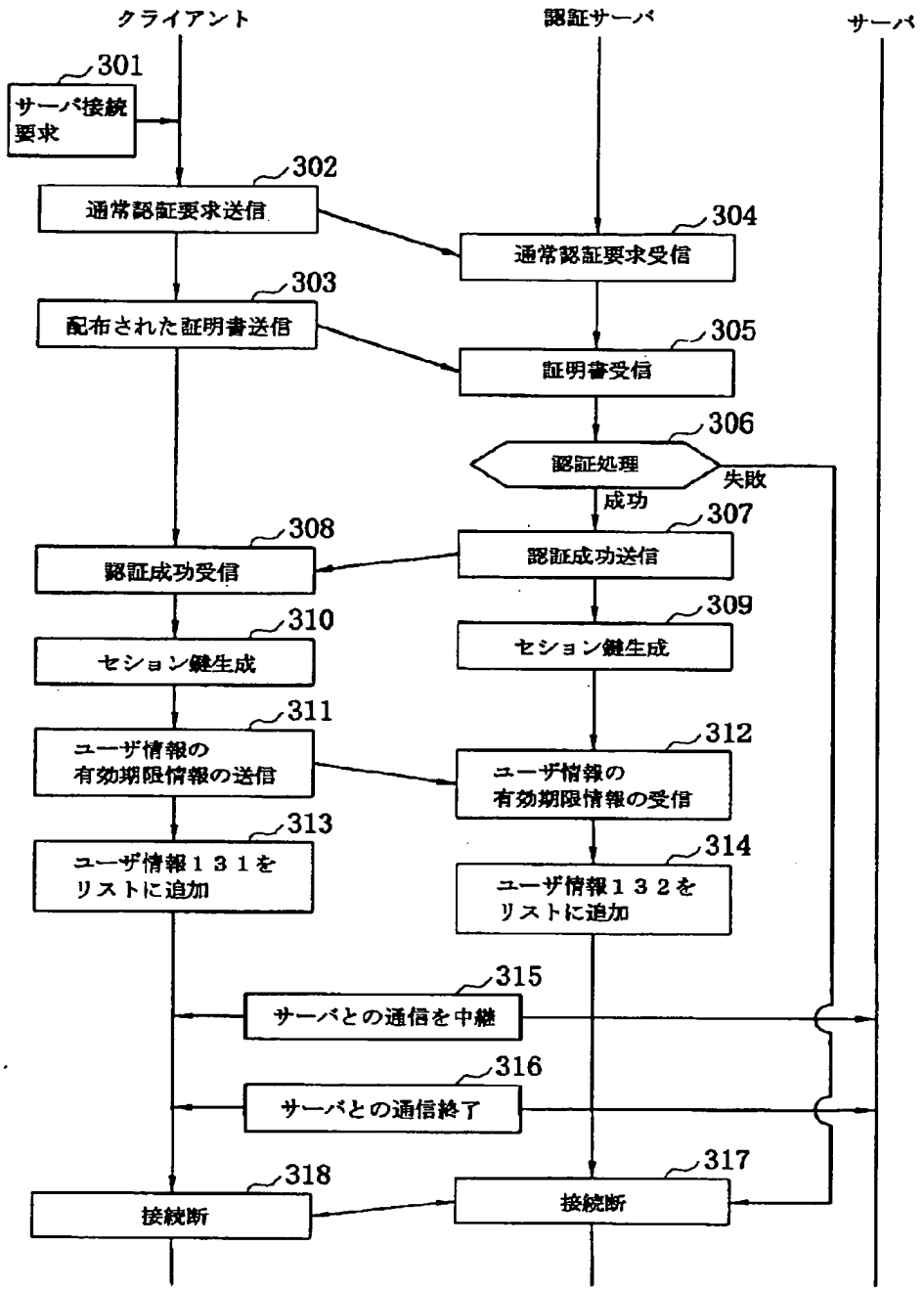


【図２】

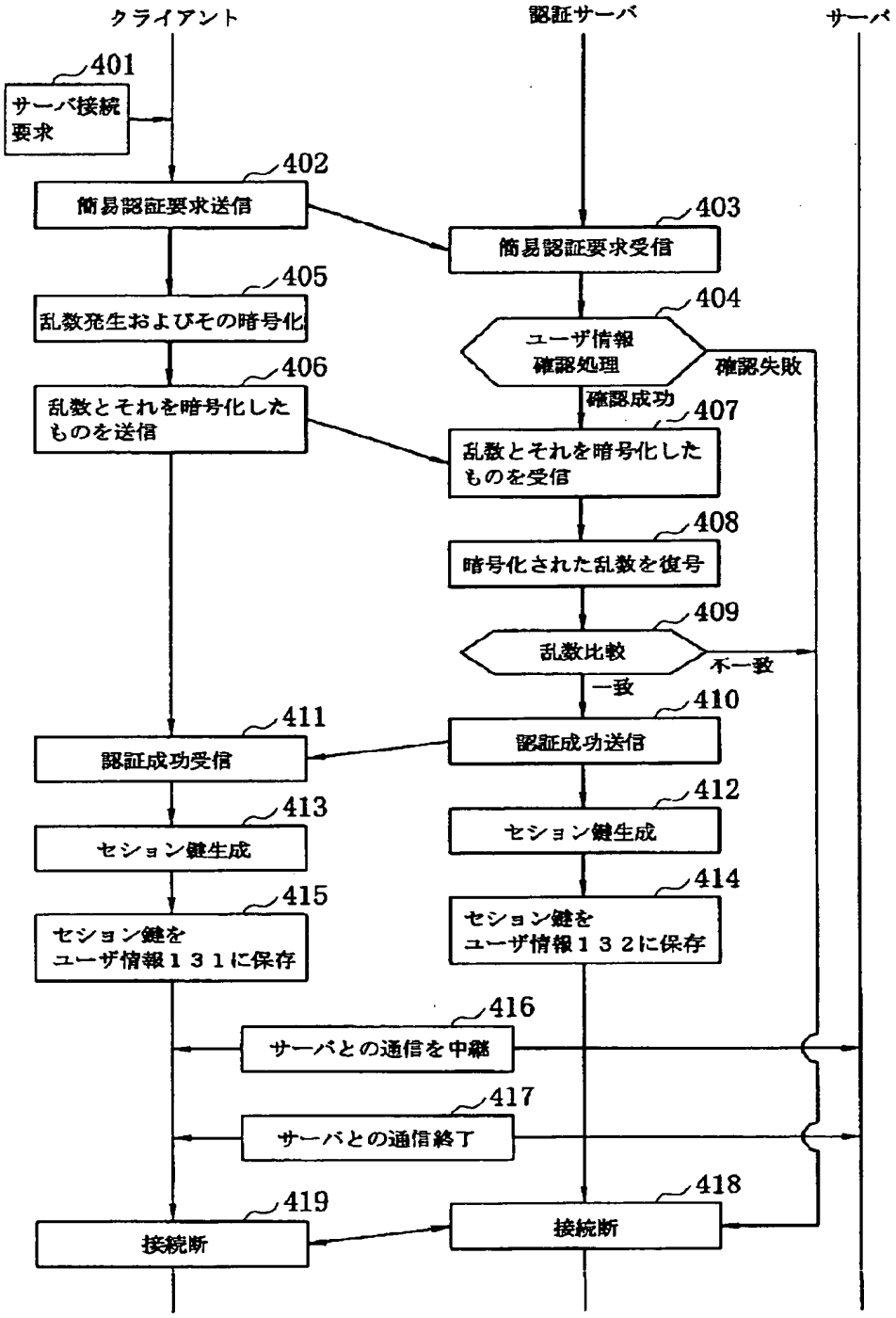


(*) クライアントのユーザ情報は接続先の、認証サーバのユーザ情報は接続元のアドレスとポート番号とする
(**) ここでフラグとは使用するコネクション数かコネクション存続時間のどちらかのメンバが有効かを示すフラグ

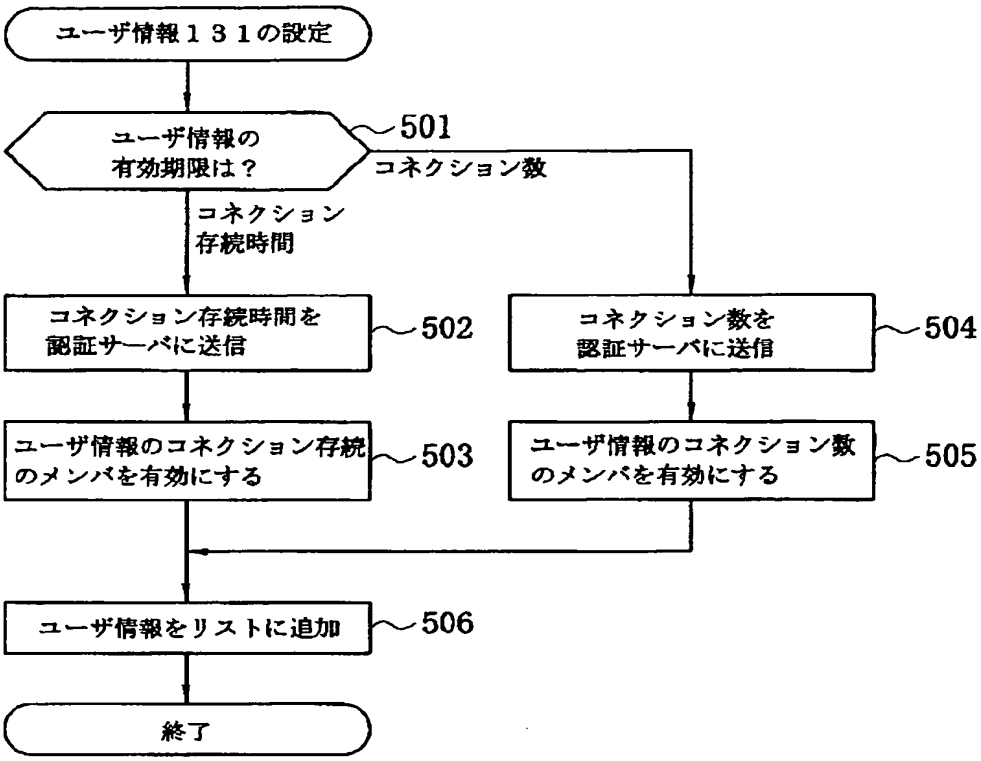
【図３】


クライアント
認証サーバ
サーバ
301
サーバ接続要求
302
通常認証要求送信
304
通常認証要求受信
303
配布された証明書送信
305
証明書受信
306
認証処理
失敗
成功
307
認証成功送信
308
認証成功受信
309
セション鍵生成
310
セション鍵生成
311
ユーザ情報の有効期限情報の送信
312
ユーザ情報の有効期限情報の受信
313
ユーザ情報１３１をリストに追加
314
ユーザ情報１３２をリストに追加
315
サーバとの通信を中継
316
サーバとの通信終了
317
接続断
318
接続断

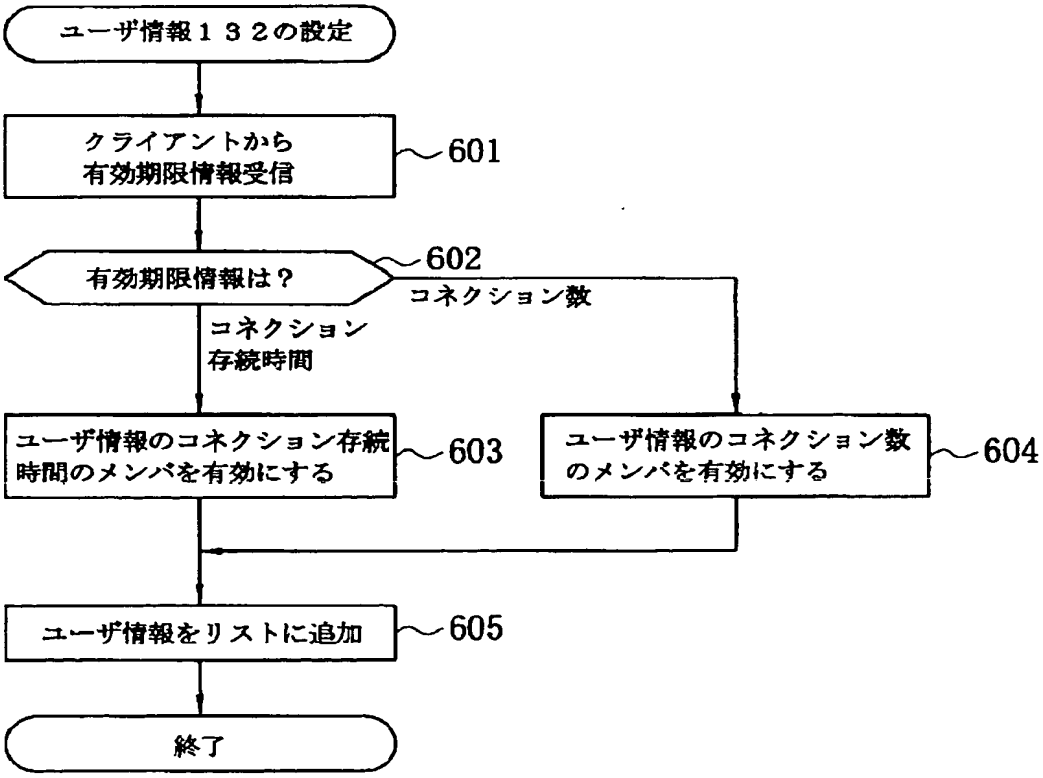
【図４】


クライアント
認証サーバ
サーバ
401
サーバ接続要求
402
簡易認証要求送信
403
簡易認証要求受信
404
ユーザ情報確認処理
確認失敗
確認成功
405
乱数発生およびその暗号化
406
乱数とそれを暗号化したものを送信
407
乱数とそれを暗号化したものを受信
408
暗号化された乱数を復号
409
乱数比較
不一致
一致
410
認証成功送信
411
認証成功受信
412
セション鍵生成
413
セション鍵生成
414
セション鍵をユーザ情報１３２に保存
415
セション鍵をユーザ情報１３１に保存
416
サーバとの通信を中継
417
サーバとの通信終了
418
接続断
419
接続断

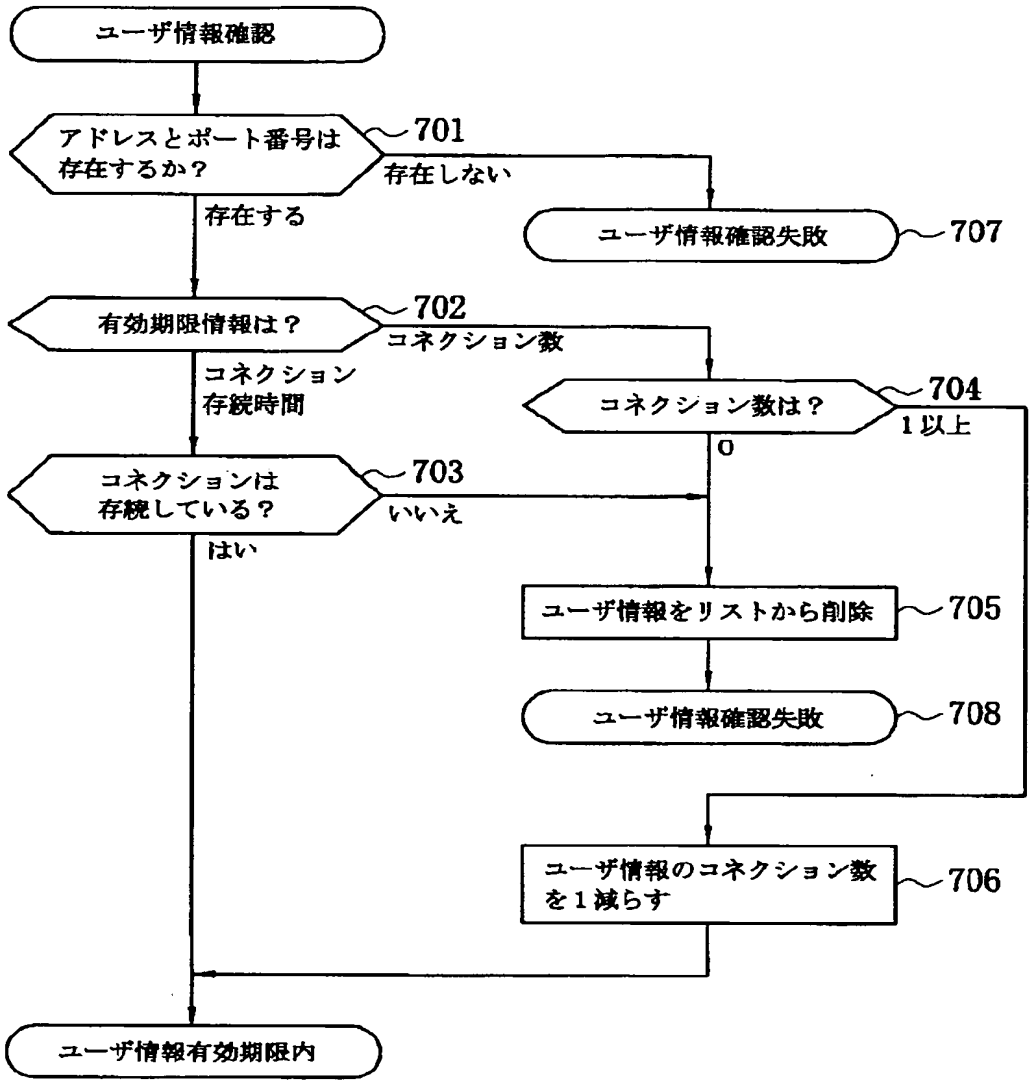
【図５】


ユーザ情報１３１の設定
ユーザ情報の
有効期限は？
501
コネクション数
コネクション
存続時間
コネクション存続時間を
認証サーバに送信
502
コネクション数を
認証サーバに送信
504
ユーザ情報のコネクション存続
のメンバを有効にする
503
ユーザ情報のコネクション数
のメンバを有効にする
505
ユーザ情報をリストに追加
506
終了

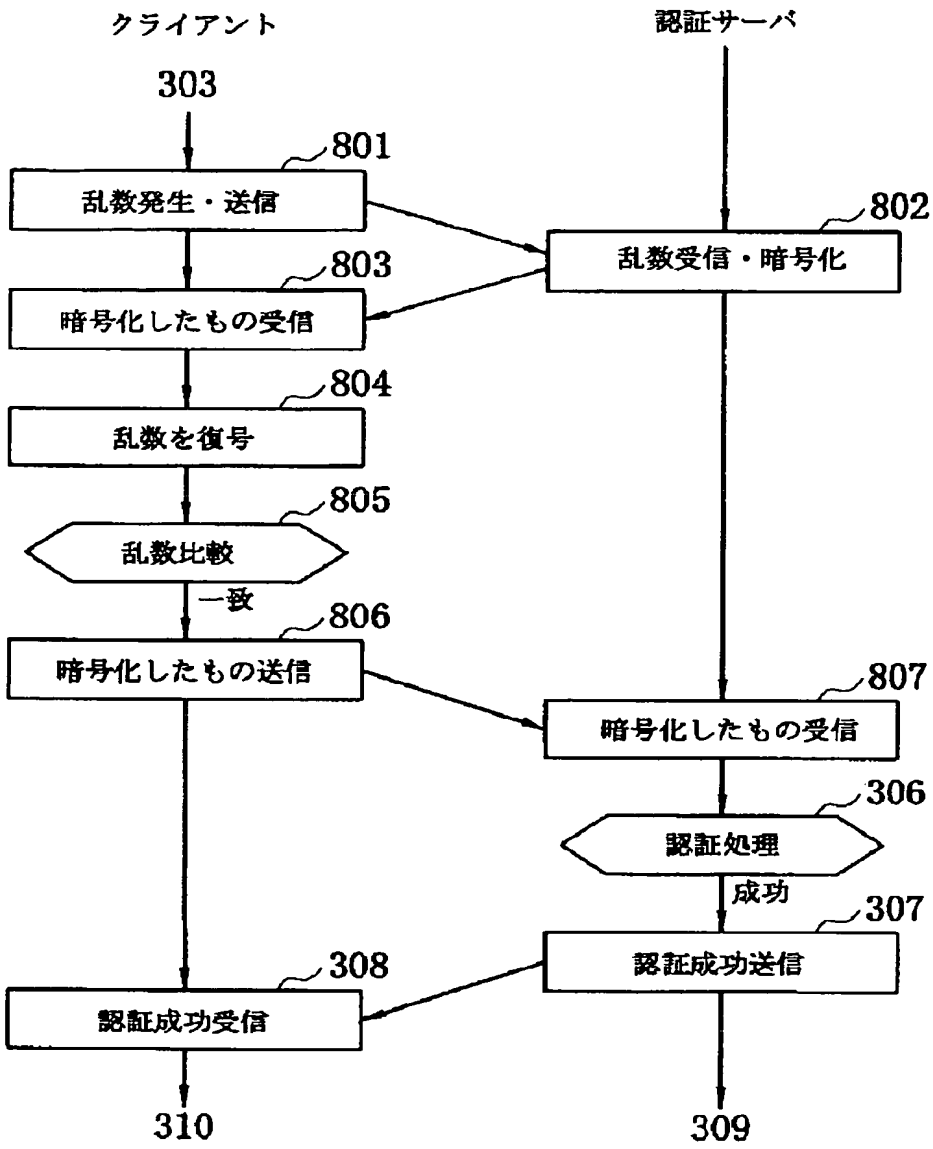
【図６】


ユーザ情報１３２の設定
クライアントから
有効期限情報受信
601
有効期限情報は？
602
コネクション数
コネクション
存続時間
ユーザ情報のコネクション存続
時間のメンバを有効にする
603
ユーザ情報のコネクション数
のメンバを有効にする
604
ユーザ情報をリストに追加
605
終了

【図７】


ユーザ情報確認
アドレスとポート番号は存在するか？
701
存在しない
存在する
ユーザ情報確認失敗
707
有効期限情報は？
702
コネクション数
コネクション存続時間
コネクション数は？
704
1以上
0
コネクションは存続している？
703
いいえ
はい
ユーザ情報をリストから削除
705
ユーザ情報確認失敗
708
ユーザ情報のコネクション数を１減らす
706
ユーザ情報有効期限内

【図８】


クライアント
認証サーバ
303
801
乱数発生・送信
802
乱数受信・暗号化
803
暗号化したもの受信
804
乱数を復号
805
乱数比較
一致
806
暗号化したもの送信
807
暗号化したもの受信
306
認証処理
成功
307
認証成功送信
308
認証成功受信
310
309

Our Ref.: OP1710-US

(Prior Art Reference)

Japanese Patent laid-Open Publication No.2000-36809A

Laid-open Date: December 2, 2000

Title of the Invention: METHOD FOR SIMPLY AUTHENTICATING USER AND RECORD MEDIUM WITH ITS PROGRAM STORED THEREIN

Application Number: 10-203654

Date of Filing: July 17, 1998

Applicant: ID No. 000005108

HITACHI LTD

Chiyoda-ku, Tokyo, Japan

Inventors: Taiji HIDAKA, Yoko SAITO

Pertinent Description ([0009]-[0017])

[0009]

Fig. 3 is a flowchart of normal authentification processing first performed according to the present invention. Fig. 4 is a flowchart showing simple authentification processing, illustrating an embodiment of the present invention. Fig. 3 and Fig. 4 are used to explain processing sequences that take place between a client, an authentification server, and a server. In Fig. 3, a server connection request is received from a user (step S301), and a client 111 sends a normal authentification request to the authentification server 112 (step S302). Then, when a certification, which has been

distributed to the client 111, is sent to the authentification server 112 (step S303), the authentification server 112 receives the certification (step S305) and then verifies the certification by comparing it against a certification which it itself has been holding (step S306). If the certifications do not match each other, then an authentification failure is notified to the client 111, and the connection with the client 111 is disconnected (step S317). By contrast, if the certifications do match, then a response indicating an authentification success is sent to the client 111 (step S307), and a session key is generated (step S309). When the authentification succeeds (step S308), the session key is generated on the client 111 side (step S310), and an address, a port number, the session key, a flag, a number of connections or a connection duration time and other such information are configured as user information 131 and sent to the authentification server 112 (step S311). Then, the user information 131 is added to a list (step S313). Referring to Fig. 5, explanation will be given regarding a flowchart with details pertaining to configuration of these settings. Fig. 3 depicts the authentification processing (step S306) in a simple fashion, but in actuality this processing also involves multiple sequences for exchanging random numbers and the like, which will be explained below using Fig. 8. Next, the authentification server 112 performs processing (steps S312, 314) to configure user information 132, which is shown in Fig. 6, based on the above-mentioned information

received, and relays the connection request from the client 111 to the server 113 (step S315). If the communications with the server 113 have ended (step S316), then the connection between the client side and the authentification server side is disconnected (steps S318, 317).

[0010]

Fig. 8 is a detailed sequence chart of the authentification processing in Fig. 3, which is performed when the normal authentification is performed. In Fig. 3, after the certification which was distributed to the client side has been sent (step S303), the processing then moves to Fig. 8, where a random number is generated and that random number is sent to the authentification server side (step S801). On the authentification server side the received random number is encoded using an encoding key which the authentification server 112 itself holds, and this is then sent to the client side (step S802). The encoded random number is then received on the client side (step S803) and the random number is decoded (step S804), and then the decoded random number is compared against the random number which the client itself has (step S805). If these random numbers do match each other, then the encoded random number is once again sent to the authentification server side (step S806). The authentification server side then receives the encoded random number (step S807), and compares it against the random number encoded in the beginning at step S802. If these do match each other, and if

the content of the certification sent from the client side is compared against the content which the authentification server 112 itself is holding and these also match each other, then the authentification is judged to be successful and the authentification success is sent to the client side (step S307). When the authentification success is received on the client side (step S308), the session key is then generated (step S310) and communications after that point are performed with the session key.

[0011]

Next, Fig. 4 is used to explain simple authentification processing in accordance with the present invention. When the server connection request is received from the user (step S401), the client 111 compares the address and the port number of the server which is the connection destination, and the above-mentioned user information 131 which has been stored. If these do match each other, then it is assumed that the authentification for connecting to this destination is complete, and a simple authentification request is sent to the authentification server 112 (step S402). On the other hand, when the authentification server 112 receives the simple authentification request from the client 111 (step S403), it then performs user information confirmation processing (step S404). Details of this confirmation processing are described below using Fig. 7. In order for the authentification server 112 to distinguish whether this is the normal authentification or the simple

authentification, the address and port number of the client which is acting as the connection source are compared against the user information 132 which the authentification server 112 is holding. If these do not match each other, then the authentification failure is notified to the client 111, and the connection with the client 111 is disconnected (step S418).

[0012]

If they do match each other, then the client 111 generates a random number and this is encoded using the session key for the user information 131 held in the client 111 (step S405). The random number and the encoded random number are then sent to the authentification server 112 (step S406). The authentification server 112 receives these (step S407), and then decodes the encoded random number using the session key of the user information 132 which it itself is holding (step S408), and then compares it against the random number which was sent to it simultaneously (step S409). If these do not match each other, then the authentification failure is sent to the client 111, and the connection to the client 111 is disconnected (step S418). If they do match, then the authentification success is sent to the client 111 (step S410). When the client 111 receives the authentification success (step S411), it then generates a new session key from the random number and the encoded random number (step S413) and stores the new session key in the user information 131 (step S415). Meanwhile, the

authentification server 112 also generates a session key at the same time as the client 111 (step S412) and stores this in the user information 132 (step S414). Then, the authentification server 112 relays the connection request from the client 111 to the server 113 (step S416). When the communications with the server 113 have ended (step S417), the connection between the client side and the authentification server side are then disconnected (steps S419, S418).

[0013]

The foregoing is the simple authentification processing of the present invention, which is performed the second time and thereafter. Comparison between the normal authentification processing shown in Fig. 3 and Fig. 8 and the simple authentification processing of the present invention shown in Fig. 4 will clarify that in the normal authentification the certification is sent and received but in the simple authentification the certification is omitted and the simple authentification request suffices whereby reducing the number of times communications are performed by 1 time. Furthermore, in the normal authentification the random number is generated on the client side, the random number itself is sent and received, and then the random numbers are compared on the client side. Thus, the random numbers must be communicated 3 times. In the simple authentification, however, the random number is generated on the client side, but the encoded random number is sent to the

authentification server and the comparison of the random numbers is performed on the authentification server side. Thus, the communication only has to be performed 1 time. Furthermore, whereas valid term information is sent and received in the normal authentification, in the simple authentification the valid term information of the user information has already been sent and received at the time when the user information confirmation processing is performed (step S404). Thus, extra communications are not necessary. Ultimately, the total number of communications decreases by 4: once in transmission of the certification, twice in the transmission to compare the random numbers, and once in the transmission of the valid term of the user information. As a result, the user authentification can be performed during a short duration of time, which also reduces the overhead for the authentification server. Moreover, when performing the simple authentification, the encoded random number is used as the subsequent session key. Therefore, the authentification can be achieved in a short duration of time without sacrificing the level of security.

[0014]

Fig. 5 is a flowchart showing user information setting processing performed at the client. Fig. 5 is used to explain a sequence of processing for setting the user information 131 on the client 111 side. In the case where the client 111 has succeeded with the normal authentification, the client 111 then petitions

the authentification server 112 for either a number of connections for it to use, or for a connection duration time to the connection destination (steps S502, S504). In the case where the client has petitioned for a number of connections for it to use, the user information remains valid until the petitioned number of connections are established (step S503). Furthermore, in the case where the client has petitioned for the connection duration time to the connection destination, the user information remains valid during the connection to the given connection destination server (step S505). After that, the address and the port number of the connection destination server, the session key that was generated, and the valid term are added to the list (step S506).

[0015]

Fig. 6 is a flowchart showing user information setting processing which takes place at the authentification server. Fig. 6 is used to explain a processing sequence for setting the user information 132 at the authentification server 112. In the case where the normal authentification has succeeded, the authentification server 112 receives the valid term information of the user information from the client 111 (step S601), and then validates the number of connections or the connection duration time in the user information 132 based on the received information (step S603, S604). After that, the address and the port number of the connection source client, the session key that was generated, and

the valid term are added to the list as the user information (step S605).

[0016]

Fig. 7 is a flowchart showing user information confirmation processing which takes place at the authentification server. Finally, Fig. 7 is used to explain a method of confirmation of the user information at the authentification server 112. In the case of the simple authentification, the authentification server 112 confirms whether or not the previously authentified user and the connection source are the same (as determined by the presence of the address and the port number) (step S701). After that, it must then be confirmed whether or not the content of the user information 132 is still valid (steps S702, S703, S704). At the authentification server 112, when performing the simple authentification, the user information is deleted from the list (step S705) if the validity has run out (step S703) or if the simple authentification has failed (step S707). In the case where the simple authentification has failed, the client 111 deletes the user information from the list.

[0017]

The present invention can be realized anywhere by converting into a program each of the operations in the simple authentification sequence chart shown in Fig. 4, the flowchart of the client's user information setting processing shown in Fig. 5, the flowchart of the authentification server's user information setting processing

shown in Fig. 6, the flowchart of the user information confirmation processing shown in Fig. 7, the normal authentification sequence chart shown in Fig. 3, and the sequence chart showing the comparison of the random numbers when performing the normal authentification as shown in Fig. 8, and then storing this converted program onto a CD-ROM or hard disk or other storage medium so that these storage media can be loaded and executed on a freely selected server and client terminal in a communications system.